Maze Solver

Project Description:

Objective:

A maze is a 2D matrix in which some cells are blocked, which is denoted as 0 and those cells are open to move that are denoted by 1.

One of the cells is the source cell from the given matrix which is on (0,0) position, from where we have to start to find the path.

And another is destination cell from the given matrix which is the (n-1, n-1) position where we have to reach. [Here the n indicates the size of matrix]

We have to find a path from the source to the destination cell without moving into any of the blocked cells.

To solve this puzzle, we first start with the source cell and move in a direction where the path is not blocked.

If taken path makes us reach to the destination then the puzzle is solved else, we have to come back and change our direction of the path which we have chosen.

We are going to implement the same logic in our code also to find the path from source to destination cell.

Algorithm

Firstly, we will make a matrix to represent the maze, and the elements of the matrix will be either 0 or 1. 0 will represent the blocked cell and 1 will represent the cells in which we can move. The matrix for the maze to solve shown below:

1 1 0 1 1

1 1 1 1 0

0 0 0 1 0

0 0 1 1 1

1 0 0 0 1

Now, we will make one more matrix of the same dimension to store the solution. Its elements will also be either 0 or 1. 1 will represent the cells in our path and rest of the cells will be 0. The matrix representing the solution is:

Here is the solution matrix which will be zero in initial stage and as we get the correct path the 0 will become 1:

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

Thus, we now have our matrices. Next, we will find a path from the source cell to the destination cell and the steps we will take are:

Check for the current cell, if it is the destination cell, then the puzzle is solved.

If not, then we will try to move downward and see if we can move in the downward cell or not (to move in a cell it must be vacant and not already present in the path).

If we can move there, then we will continue with the path taken to the next downward cell.

If not, we will try to move to the rightward cell. And if it is blocked or taken, we will move upward.

Similarly, if we can't move up as well, we will simply move to the left cell.

If none of the four moves (down, right, up, or left) are possible, we will simply move back and change our current path (backtracking).

And the after in solution matrix we will get our path for the maze input matrix that is shown in picture below.

1 0 0 0 0

1 1 1 1 0

0 0 0 1 0

0 0 0 1 1

0 0 0 0 1

Execution Step’s:

To run this Project in your device first install it from main.exe file then edit your desired input in the input.txt file

and then run (run.bat) file and you will get your output in output.txt file with time and date history.

Modules Used

1. argparse - For creating a Command line interface

2. datetime - For date and time history.